AMENDMENTS TO THE SPECIFICATION

Page 1, lines 7-13

Hardware verification is currently the bottleneck and the most expensive task in the design of a semiconductor integrated circuit. Model checking is a method of formal verification that is gaining in popularity for this purpose. The method is described generally by Clarke et l. in *Model Checking* (MIT Press, 1999)~~, which is incorporated herein by reference~~.

Page 2, lines 1-11

Model checking is preferably carried out automatically by a symbolic model checking program, such as SMV, as described, for example, by McMillan in *Symbolic Model Checking* (Kluwer Academic Publishers, 1993), ~~which is incorporated herein by reference~~. A number of practical model checking tools are available, among them RuleBase, developed by IBM, which is described by Beer et al. in "RuleBase: an Industry-Oriented Form Verification Tool," in *Proceedings of the Design Automation Conference DAC'96* (Las Vegas, Nevada, 1996)~~, which is incorporated herein by reference~~.

Page 2, line 22 - Page 3, Line 9

Coverage metrics have been applied in various fields of simulation-based verification in order to measure and improve the completeness with which a given simulation tool represents the actual behavior of a target system. An application of such a metric to model checking is described by Hoskote et al., in "Coverage Estimation for Symbolic Model Checking," in *Proceedings of the Design Automation Conference DAC'99* (IEEE Computer Society Press, 199)~~, which is incorporated herein by reference~~. ~~The authors present~~ This publication presents a method for estimating whether a set of properties is sufficient to cover all possible states of a model. ~~They note~~ It notes, however, that ~~their~~ the disclosed method cannot point out functionality that may be missing in the model, nor can it ensure that all possible paths

between the states are covered. ~~They indicate~~ The publication also indicates that "path coverage would be an ideal coverage metric because it can provide coverage of actual executions of the circuit over time." The ~~authors consider~~ publication considers that by comparison with state coverage, "path coverage is a much more intractable problem."

Page 15, lines 3-13

The above formulas correspond to the proeprties $\Phi 0$, $\Phi 1$, ..., $\Phi 8$ listed symbolically below in Table II, which are a complete specification of arbiter 40 written in the form of a safety formula ψ in Universal Computation Tree Logic (known as ACTL). ACTL is a branching-time temporal logic. It is described in detail by Grumberg et al. in "Model Checking and Modular Verification," in *ACM ransactions on Programming Language and Systems* 16(3) (1994), pp. 843-871, ~~which is incorporated herein by reference~~. As the variable "robin" is not observable, it does not appear explicitly in the properties in Table II.

Page 21, lines 20-28

Preferably, for efficient computation, S_i, S_t, R_i, R_t, L_i, L_t and SIM_j are all represented as Ordered Binary Decision Diagrams (OBDDs). This type of representation, using connected, directed, acyclic graphs, is hown in the art of model checking. The use of OBDDs in this regard is described, for example, by McMillan in *Symbolic Model Checking* (Kluwer Academic Press, Norwell, Massachusetts, 1993), ~~which is incorporated herein by reference~~.